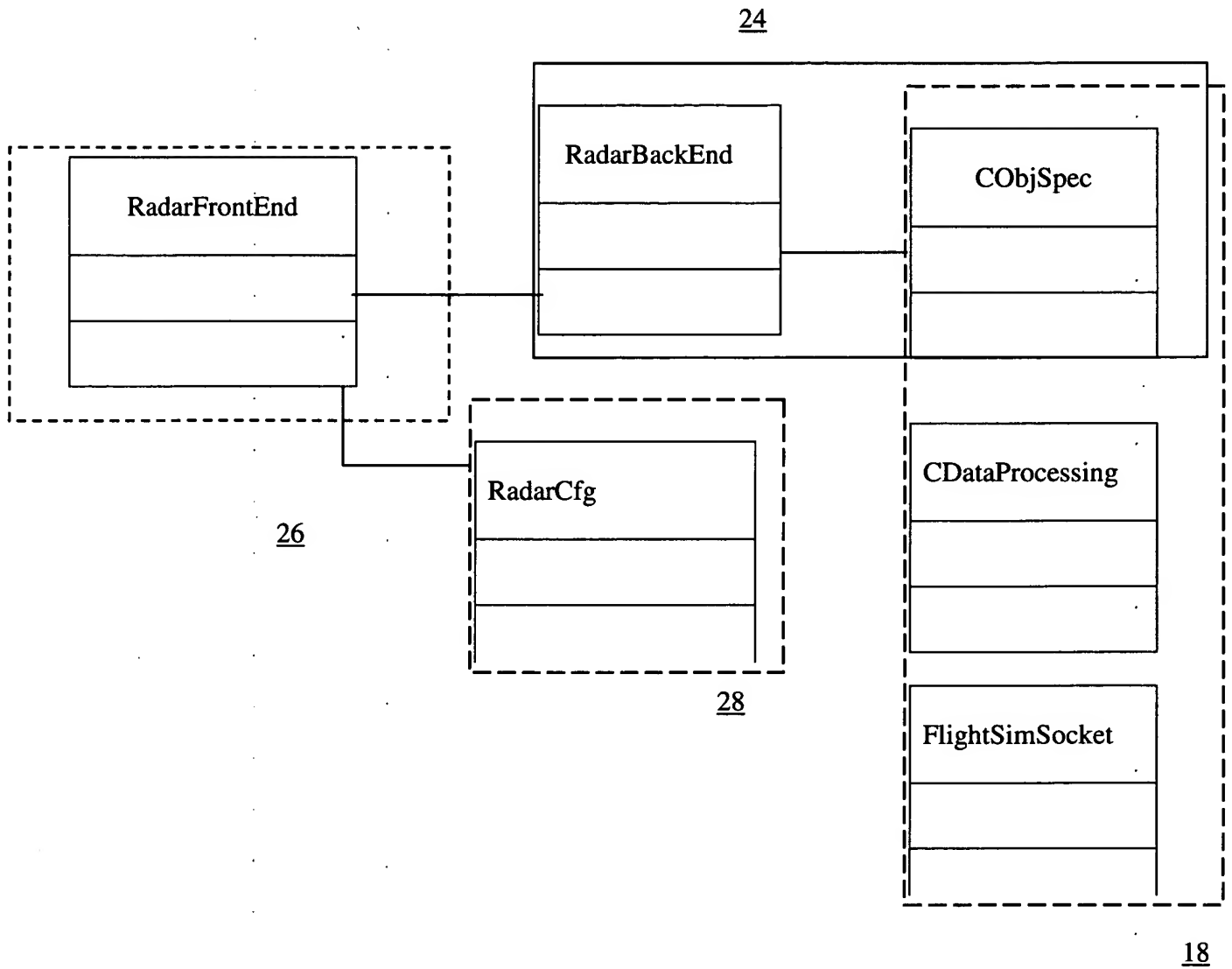




Appl. No 10/749,361
Amdt. Dated September 1, 2005
Reply to Office action of June 16, 2005
Replacement Sheet

FIG 4





Appl. No 10/749,361
Amdt. Dated September 1, 2005
Reply to Office action of June 16, 2005
Replacement Sheet

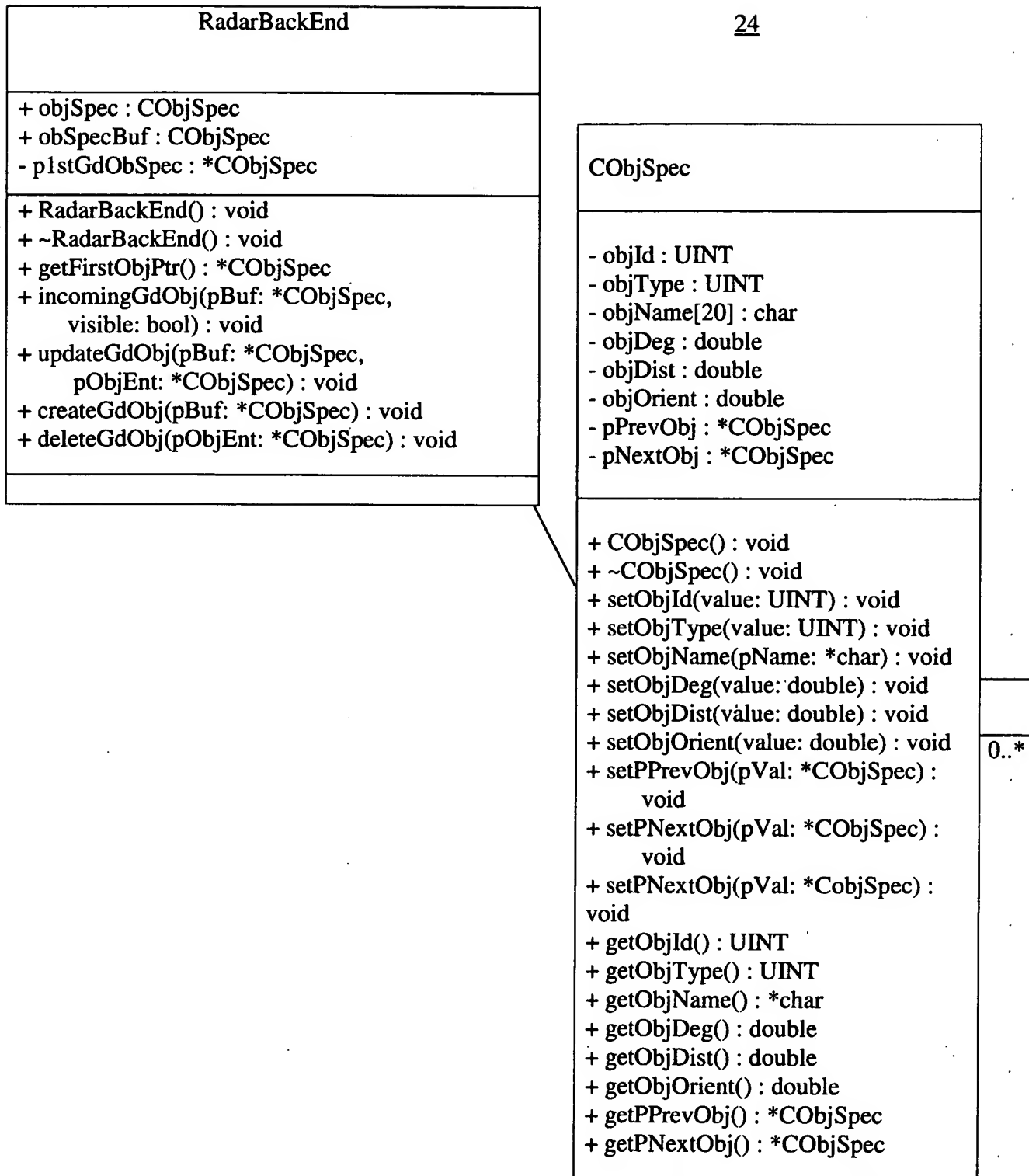
FIG 4

RadarFrontEnd
<ul style="list-style-type: none"> - bRotate : bool - bClearDisp : bool - bStopRendering : bool - bStandby : bool - iLinearSize : GLuint - cxCenter : float - cyCenter : float - lSweepAngle : float - lAlphaFade : float - lSweepIncrement : float - lRange : float - lGainFactor : float - hpTextures[10] : GLuint - lpSweepTexture[128][4] : GLfloat - uipRenderTexture[65536][3] : GLuint - pRadarBackEnd : *RadarBackEnd - pRadarCfg : *RadarCfg - pFirstNtt : *CobjSpec
<ul style="list-style-type: none"> + RadarFrontEnd(pConfig: RadarCfg, pBackEnd: RadarBackEnd, cxWidth: GLint, cyHeight: GLint) : void + ~RadarFrontEnd() : void + renderScene() : void + updateParameters() : void + pauseRendering() : void + continueRendering() : void + getHeloYaw() : void - orthoMode(xLeft: GLint, xRight: GLint, yBottom: GLint, yTop: GLint) : void - perspectiveMode() : void - createSweep(uiTextureID: GLuint, lxCenter: GLfloat, lyCenter: GLfloat, lzCenter: GLfloat, lxWidth: GLfloat, lyLength: GLfloat, lzHeight: GLfloat) : void - createTexture(uiTextureID: GLuint) : void - renderMotionBlur(uiTextureID: GLuint) : void - renderHeloSymbol() : void - drawBlip() : void

26



FIG 4





Appl. No 10/749,361
Amdt. Dated September 1, 2005
Reply to Office action of June 16, 2005
Replacement Sheet

FIG 4

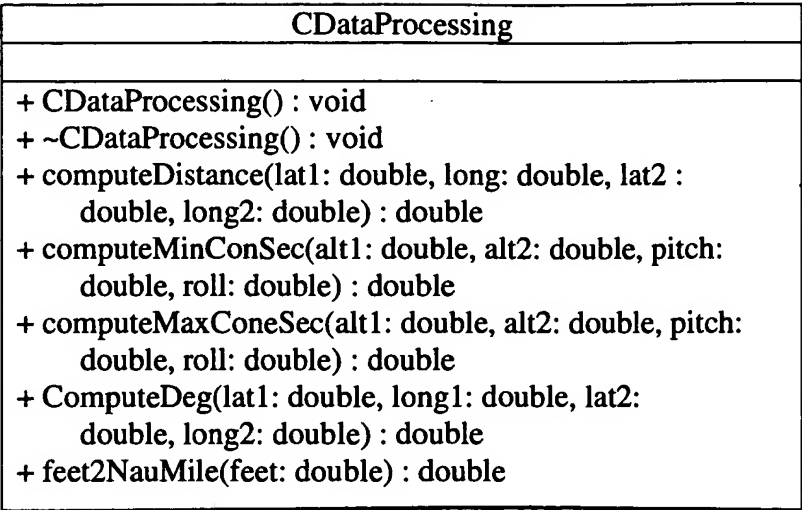
28

RadarCfg
<ul style="list-style-type: none">- iModeSpeed : UINT- iRcvrGain : UINT- iStab : UINT- iEraseGPI : UINT- iPersist : UINT- iRange : UINT- cxHelo : UINT- cyHelo : UINT- cyHeloOffset : UINT
<ul style="list-style-type: none">+ RadarCfg(rModeSpeed: UINT, rRcvrGain: UINT, rStab: int, rEraseGPI: int, rPersist: UINT, rRange:UINT, rXPos: UINT, rYPos: UINT, rYOffset: UINT) : void+ setModeSpeed(rParam: UINT) : void+ setRcvrGain(rParam: UINT) : void+ setStab(rParam: UINT) : void+ setEraseGPI(rParam: UINT) : void+ setPersist(rParam: int) : void+ setRange(rParam: UINT) : void+ setHeloXPos(rParam: UINT) : void+ setHeloYPos(rParam: UINT) : void+ setHeloYOffset(rParam: UINT) : void+ getModeSpeed() : UINT+ getRcvrGain() : UINT+ getStab() : UINT+ getEraseGPI() : UINT+ getPersist() : UINT+ getRange() : UINT+ getHeloXPos() : UINT+ getHeloYPos() : UINT+ ~RadarCfg() : void



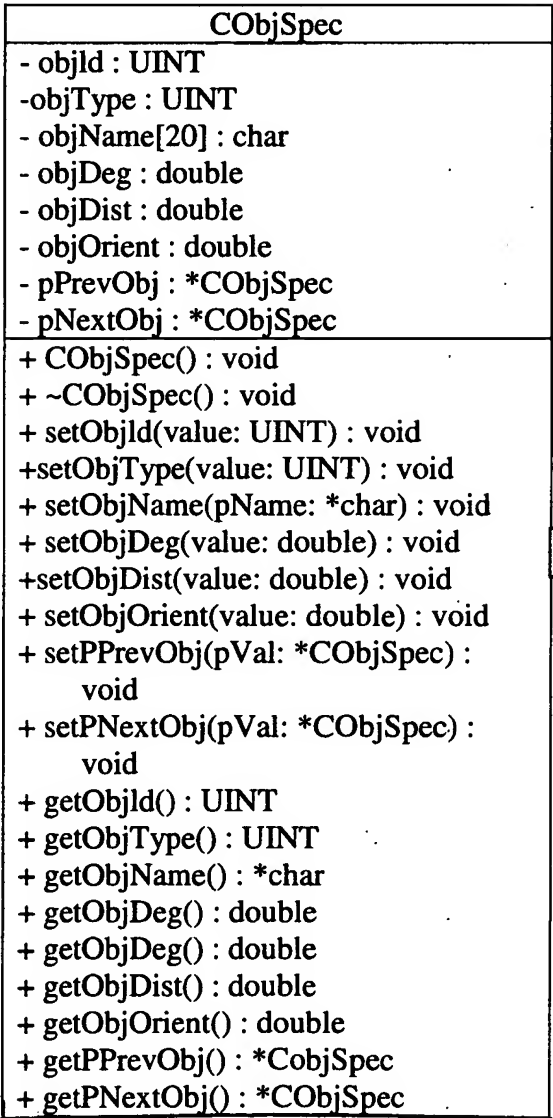
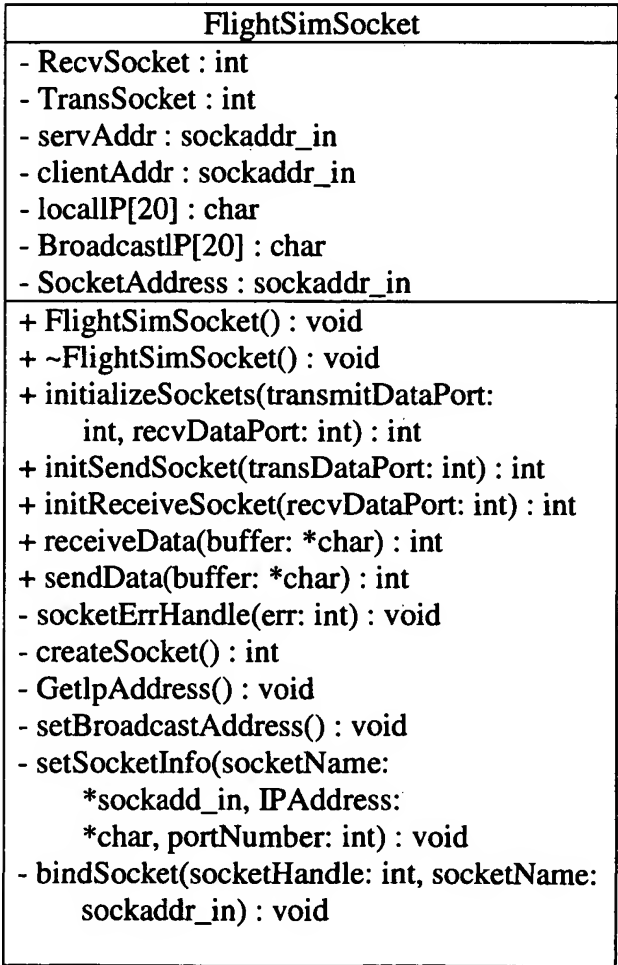
FIG 4

18



RADAR beam propagation model and calculations are independent from the rest of the software.

Network thread implementation receives data and executes separate from the rest of the software.



0..*